Algorithmic Number Theory

Course 11

Jean-Sébastien Coron

Université du Luxembourg

December 5, 2009

Summary

- Algorithmic number theory.
  - Generators of $\mathbb{Z}_p$
  - Discrete logarithm and applications.

- Definitions
  - A group G is *finite* if $|G|$ is finite. The number of elements in a finite group is called its *order*.
  - A group G is *cyclic* if there is an element $g \in G$ such that for each $h \in G$ there is an integer i such that $h = g^i$. Such an element g is called a generator of G.
  - Let G be a finite group and $a \in G$. The *order* of a is definded to be the least positive integer t such that $a^t = 1$.

- Facts
  - Let G be finite group and $a \in G$. The order of a divides the order of G.
  - Let G be a cyclic group of order n and $d|n$, then G has exactly $\phi(d)$ elements of order d. In particular, G has $\phi(n)$ generators.

Properties of $\mathbb{Z}_n^*$

- Definition of $\mathbb{Z}_n^*$
  - The set $\mathbb{Z}_n^*$ is the set of integers modulo n which are invertible modulo n.
  - The set $\mathbb{Z}_n^*$ is a group of order $\phi(n)$ for the operation of multiplication modulo n.
- Properties
  - $\mathbb{Z}_p^*$ for prime p is a cyclic group of order $p-1$.
  - There exists a generator $g \in \mathbb{Z}_p^*$ such that for all $\alpha \in \mathbb{Z}_p^*$, α can be written uniquely as $\alpha = g^x \mod p$ for $0 \leq x < p-1$.
  - The integer x is called the *discrete logarithm* of α to the base g, and denoted $\log_g \alpha$.

Finding a generator of $\mathbb{Z}_p^*$

- Finding a generator of $\mathbb{Z}_p^*$ for prime p.
  - The factorization of $p-1$ is needed. Otherwise, no efficient algorithm is known.
  - Factoring is hard, but it is possible to generate p such that the factorization of $p-1$ is known.
- Generator of $\mathbb{Z}_p^*$
  - $g \in \mathbb{Z}_p^*$ is a generator of $\mathbb{Z}_p^*$ if and only if $g^{(p-1)/q} \neq 1 \mod p$ for each prime factor q of $p-1$.
  - There are $\phi(p-1)$ generators of $\mathbb{Z}_p^*$

Finding a generator

- Let $q_1, \ldots q_r$ be the prime factors of $p-1$
  - 1) Generate a random $g \in \mathbb{Z}_p^*$
  - 2) For $i = 1$ to r do
    - Compute $\alpha_i = g^{(p-1)/q_i} \mod p$
    - If $\alpha_i = 1 \mod p$, go back to step 1.
  - 3) Output g as a generator of $\mathbb{Z}_p^*$
- Complexity:
  - There are $\phi(p-1)$ generators of $\mathbb{Z}_p^*$.
  - A random $g \in \mathbb{Z}_p^*$ is a generator with probability $\phi(p-1)/(p-1)$.
  - If $p - 1 = 2 \cdot q$ for prime q, then $\phi(p-1) = q - 1$ and this probability is $\simeq 1/2$.

Generating p and q

- Goal: generate p such that $p - 1 = 2 \cdot q$ for prime q.
  - Generate a random prime p.
  - Test if $q = (p - 1)/2$ is prime. Otherwise, generate another p.

Discrete logarithm

- Let g be a generator of $\mathbb{Z}_p^*$
  - For all $a \in \mathbb{Z}_p^*$, a can be written uniquely as $a = g^x \mod p$ for $0 \leq x < p - 1$.
  - The integer x is called the *discrete logarithm* of a to the base g, and denoted $\log_g a$.
- Computing discrete logarithms in $\mathbb{Z}_p^*$
  - Hard problem: no efficient algorithm is known for large p.
  - Brute force: enumerate all possible x. Complexity $\mathcal{O}(p)$.
  - Baby step/giant step method: complexity $\mathcal{O}(\sqrt{p})$.

Diffie-Hellman protocol

- Enables Alice and Bob to establish a shared secret key that nobody else can compute, without having talked to each other before.
- Key generation
  - Let p a prime integer, and let g be a generator of $\mathbb{Z}_p^*$. p and g are public.
  - Alice generates a random x and publishes $X = g^x \mod p$. She keeps x secret.
  - Bob generates a random y and publishes $Y = g^y \mod p$. He keeps y secret.

Diffie-Hellman protocol

- Key establishment
  - Alice sends X to Bob. Bob sends Y to Alice.
  - Alice computes $K_a = Y^x \mod p$
  - Bob computes $K_b = X^y \mod p$

$$K_a = Y^x = (g^y)^x = g^{xy} = (g^x)^y = X^y = K_b$$

- Alice and Bob now share the same key $K = K_a = K_b$
  - Without knowing x or y, the adversary is unable to compute K.
  - Computing g^{xy} from g^x and g^y is called the *Diffie-Hellman problem*, for which no efficient algorithm is known.